2178

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of: Charisius et al.

Group Art Unit: 2178

Application No: 09/944,696

Examiner: Stork, Kyle R

Filed: 08/31/2001

For: **METHODS AND SYSTEMS FOR ANIMATING A WORKFLOW AND A PROJECT PLAN**

APPEAL BRIEF FOR APPLICATION NO. 09/944,696

Board of Patent Appeals and Interferences
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Sirs:

Transmittal of Appeal Brief

This Appeal Brief is being transmitted in triplicate in this application with respect to the Notice of Appeal filed on August 31, 2005. The application is on behalf of a large entity. The required fee of $500.00 for filing the Appeal Brief is attached along with $120 for the fee for a one-month extension of time to file this brief. Please provide a one-month Extension of Time up to and including December 2, 2005 to file the Appeal Brief. Commissioner is authorized to charge any additional fees that may be required to Deposit Account 501923.

12/02/2005 HDEMESS1 00000075 09944696

01 FC:1402 500.00 OP
02 FC:1251 120.00 OP

APPELLANT'S BRIEF

1. Real Party in Interest

The real party in interest in this appeal is Borland Software Corporation, the parent corporation of Togethersoft, the assignee of record.

2. Related Appeals and Interferences

There are no related Appeals and Interferences.

3. Status of the Claims

Claims 1, 2, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 32, 33, 34, 35, 36, and 37 are withdrawn. Claims 3-17, 28-31, and 38-44 are pending. Claims 3, 13, 28, 38, 41, and 44 are independent claims.

4. Status of Amendments

No claims were amended subsequent the Examiner's Final Rejection.

5. Summary of Claimed Subject Matter

The present application discloses a data processing system, a computer-readable medium and a method for integrating a business process or workflow with a project plan. More particularly, the invention relates to a method and system for creating and activating a project plan, and animating the activation of the project plan. The data processing system has versions of a plan, wherein each version reflects an instance in an edit history of the plan. For the purpose of the present application, a plan represents an instance of the workflow. While most plans can

be described in a printed document, a document describing a plan is not the plan itself. For example, a plan can be viewed as a working schedule for a project to produce a product or artifact, such as a computer, bicycle, or software build, for an enterprise.

In operation, the processing system executes instructions invoking the method stored on the computer readable medium. The method includes steps of storing indications of the plan's versions and displaying the versions of the plan in a sequential manner to simulate animation of the plan's edit history. In other words, the system shows an animation graphic with the various versions in sequence.

Versions of a plan may be displayed in a visually distinctive manner as a function of frequency of change of the edits. For example, areas of a plan that undergo the most change may be visually distinct (e.g., red or bolded or tagged with a number associated with the frequency of changes). Please see paragraph [0186] of the application. This feature allows a workflow manager to immediately distinguish between versions of a plan that have been minimally edited and those that have been frequently edited. In knowing this piece of information, the workflow manager can identify trouble spots in the workflow.

6. Grounds of Rejection to be Reviewed on Appeal

The four grounds of rejection are:

1. Claims 3-10, 13-15, 38, and 41 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Boden et al. (U.S. 5,930,512, hereafter Boden) in view of Nauckhoff (U.S. 5,893,128) and in further view of Microsoft® Word 2000. Apparently, claim 28 is also subject to this rejection – see Office Action of

June 1, 2005, page 7. Claims 3-10, 13-15, 28, 38, and 41 can be grouped together.

2. Claims 11, 16, 30, 39, and 42 stand rejected under 35 U.S.C. 103(a) as being unpatenable over Boden, Nauckhoff, and Word and in view of ls (2000, http://polyglotman.sourceforge.net/sgi-ls.l.html found on page 7, line 2). Claims 11, 16, 30, 39, and 42 can be grouped together.

3. Claims 12, 17, 40, and 43 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Boden, Nauckhoff, and Word and further in view of Kumashiro (US 6,240,395). Claims 12, 17, 40, and 43 can be grouped together.

4. Claim 29 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Boden, Nauckhoff, and Word in further view of Garofalakis et al. (US 5,845,279). Claim 29 stands by itself.

7. Argument

7.1: Claims 3-10, 13-15, 28, 38, and 41 are patentable over Boden et al. (U.S. 5,930,512, hereafter Boden) in further view of Nauckhoff (U.S. 5,893,128) and in further view of Microsoft® Word 2000.

The Present Invention Is Not Obvious Over The Cited References

A claimed invention may be found to have been obvious "if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains." 35 U.S.C. § 103(a). Moreover, the Federal Circuit

has ruled on numerous occasions that a holding of "obviousness" requires some motivation, suggestion or teaching within the cited references that would lead one skilled in the art to modify the cited reference or references as claimed by applicant. See, for example, In re Kotzab, 217 F3d 1365, 55 USPQ2d 1313 (Fed Cir. 2000):

> "Most if not all inventions arise from a combination of old elements. See In re Rouffet, 149 F.3d 1350, 1357, 47 USPQ2d 1453, 1457 (Fed. Cir. 1998). Thus, every element of a claimed invention may often be found in the prior art. However, identification in the prior art of each individual part claimed is insufficient to defeat patentability of the whole claimed invention. Rather, to establish obviousness based on a combination of the elements disclosed in the prior art, there must be some motivation, suggestion or teaching of the desirability of making the specific combination that was made by the applicant. See In re Dance, 160 F.3d 1339, 1343, 48 USPQ2d 1635, 1637 (Fed. Cir. 1998); In re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984). Even when obviousness is based on a single prior art reference, there must be a showing of a suggestion or motivation to modify the teachings of that reference. See B.F. Goodrich Co. v. Aircraft Breaking Sys. Corp., 72 F.3d 1577, 1582, 37 USPQ2d 1314, 1318 (Fed. Cir. 1996)."

The Examiner improperly holds the position that Microsoft's Word program displays a frequency of change in edit history. The Examiner points to pages 1-2 and Figure 1 of Microsoft® Word 2000, "About adding comments and keeping track of changes," 1999, to support his position. However, after carefully reviewing the Microsoft® Word 2000 reference cited by the Examiner, counsel cannot find in the cited reference where the Word program displays a frequency change in edit history. The examiner's premise for the rejection is in error, so the rejection is in error.

Edit highlights resulting from the use of the invention as claimed in these claims are displayed as a function of edit frequency, whereas the cited Microsoft ® Word 2000 reference highlights edits without regard for edit frequency. As a result, and in contrast with applicant's claims, a user of Microsoft® Word 2000 cannot quickly and efficiently determine the frequency of change in the edit history of a plan.

Claims 3 and 13 and their dependents recite that the editing is made to plans and workflow, respectively. The ability to quickly determine the frequency of edit history allows a user (i.e., a manager) to quickly and accurately determine if a process has undergone heavy editing. In contrast, Microsoft® Word 2000 is directed towards the preparation of printed material. A document generated using Microsoft® Word 2000 is not a plan. A Microsoft® Word 2000 generated document generally contains text and/or graphics for a publication, and is not suitable for animating a workflow or process plan. As a result, there is no motivation to modify Microsoft® Word 2000 to display highlights as a function frequency of change in edit history. Moreover, Microsoft® Word 2000 would gain no apparent benefit by adding such a feature. Nor is there a suggestion, teaching or motivation for combining Boden and Nauckhoff with Microsoft® Word 2000 to come up with a method of displaying portions of a plan in a visually distinctive manner as a function of frequency of change of the edits.

Therefore, Applicants respectfully request that the rejections of claims 3-10, 13-15, 28, 38, and 41 be reversed.

7.2: Claims 11, 16, 30, 39, and 42 are patentable over Boden, Nauckhoff, and Word and in view of ls (2000, found on page 7, line 2).

Claims 11, 16, 30, 39 and 42 recite that the versions of a plan can be displayed in reverse order. The independent claims from which these claims depend call for displaying the versions of the plan in a sequential manner to simulate animation of the plans edit history. Neither Boden, Nauckoff, Microsoft's Word program nor ls (2000 found on page 7, line2) disclose versions of a plan displayed in reverse order.

The examiner's final rejection admits that he needs to go to ls to find a teaching of reverse order. That teaching is as follows:

DESCRIPTION

For each directory argument, *ls* lists the contents of the directory; for each file argument, *ls* repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format is to list one entry per line, the **-C** and **-x** options enable multi-column formats, and the **-m** option enables stream output format. In order to determine output formats for the **-C**, **-x**, and **-m** options, *ls* uses an environment variable, **COLUMNS**, to determine the number of character positions available on one output line. If this variable is not set, the terminfo(4) database is used to determine the number of columns, based on the environment variable **TERM**. If this information cannot be obtained, 80 columns are assumed.

The *ls* command has the following options:

-R
Recursively list subdirectories encountered.

-a
List all entries, including those that begin with a dot (.), which are normally not listed.

.....

-o
The same as **-l**, except that the group is not printed.

-g
The same as **-l**, except that the owner is not printed.

-r
Reverse the order of sort to get reverse alphabetic or oldest first as appropriate.

The examiner makes no explanation as to how this programmer-level explanation of the parameters of an undefined code in an unexplained environment has anything to do with applicant's invention. It certainly provides no motivation to display the versions of a plan or

workflow in reverse order. At most, if the programming environment is right, the reference may show HOW to do it, but it says nothing about WHY to do it. The motivation to combine required to sustain an obviousness rejection is absent. Therefore, Applicants respectfully request that the rejections of claims 11, 16, 30, 39, and 42 be reversed.

7.3: Claims 12, 17, 40, and 43 are patentable over Boden, Nauckhoff, and Word and further in view of Kumashiro (US 6240395, patent 2001, file 1998).

Claims 12, 17, 40, and 43 pertain to the displaying of Gantt charts and flow diagrams that make up some of the visual components of the versions of the plan. For example, edits to Gantt charts or flow diagrams will be highlighted with a distinctive indicator such as a tag having a number associated with the frequency of changes made to a particular element of a Gantt chart or flow diagram.

Neither Boden, Nauckoff, Microsoft's Word 2000 program disclose Gantt charts or flow diagrams. Kumashiro discloses Gantt charts, but does not disclose edits that are highlighted with a distinctive indicator such as a tag having a number associated with the frequency of changes made to a particular element of a Gantt chart or flow diagram. Nor does Kumashiro provide a motivation to do so. Therefore, Applicants respectfully request that the rejections of claims 12, 17, 40, and 43 be reversed.

7.4: Claim 29 is patentable over Boden, Nauckhoff, and Word in further view of Garofalakis et al.

Claim 29 is dependent upon claim 28, which discloses a computer-readable medium containing instructions for controlling a data processing system to perform a method having

steps of retrieving a plurality of plans generated from a workflow, and displaying each of the plans in a sequential manner to simulate the generation of the plans from the workflow. The displaying of the plans is visually distinctive as a function of frequency of the change in the plans. Neither Boden, Nauckhoff, Word nor Garofalakis et al suggest, teach or motivate alone or in combination a computer-readable medium containing instructions for controlling a data processing system to perform a method having steps of retrieving a plurality of plans generated from a workflow, and displaying each of the plans in a sequential manner to simulate the generation of the plans from the workflow, wherein the displaying of the plans is visually distinctive as a function of frequency of the change in the plans.

Garofakalis is directed to continuous-media-on-demand ("CMOD") services and, more specifically, to systems and methods for increasing the performance of databases that provide CMOD services (so-called "continuous media databases"). The continuous media that Garofakalis refer to are video and audio, such as movies or other programming available as Enhanced Pay-Per-View ("EPPV"), video-on-demand ("VOD"), on-line tutorials and interactive television. Garoafakalis disclose methods an apparatus for retrieving portions of files from memory and sending them down media-delivery datapipes like optical fibers as efficiently as possible. Garofakalis does not say anything to suggest that one or ordinary skill in the art should modify any of the primary references to arrive at applicant's invention. As a result, Applicants respectfully request that the rejection of claim 29 be reversed.

7.5: Conclusion

The Examiner's rejections of Claims 3-17, 28-31, and 38-44 should be reversed.

8. Claims Appendix

Listing of Appealed Claims

3. A method in a data processing system having versions of a plan, each reflecting an instance in an edit history, the method comprising the steps of:

storing indications of the versions of the plan; and

displaying the versions of the plan in a sequential manner to simulate animation of the edit history;

wherein said displaying is visually distinctive as a function of frequency of change in the edit history.

4. The method of claim 3, wherein the step of storing indications of the versions of the plan comprises the steps of:

storing versions of a task of the plan; and

creating a link from the plan to the versions of the task.

5. The method of claim 3, wherein the versions of the plan reflect an activation of the plan.

6. The method of claim 5, wherein the plan comprises a plurality of tasks, and the indications of the versions of the plan comprise the states of the tasks.

7. The method of claim 6, wherein the state comprises an unexecuted state.

8. The method of claim 6, wherein the state comprises an executing state.

9. The method of claim 6, wherein the state comprises an executed state.

10. The method of claim 3, wherein the indications of versions of the plan reflect a modification to the plan.

11. The method of claim 3, wherein the versions of the plan are displayed in reverse order.

12. The method of claim 3, wherein the display comprises a Gantt chart.

13. A method in a data processing system having versions of a workflow, each reflecting an instance in an edit history, the method comprising the steps of:

storing indications of the versions of the workflow; and

displaying the versions of the workflow in a sequential manner to simulate animation of the edit history; wherein said displaying is visually distinctive as a function of frequency of change in the edit history.

14. The method of claim 13, wherein the step of storing indications of the versions of the workflow comprises the steps of:

storing versions of an activity of the workflow; and

creating a link from the workflow to the versions of the activity.

15. The method of claim 13 wherein the indications of versions of the workflow reflect a modification to the workflow.

16. The method of claim 13 wherein the versions of the workflow are displayed in reverse order.

17. The method of claim 13 wherein the display comprises a flow diagram.

28. A computer-readable medium containing instructions for controlling a data processing system to perform a method, the method comprising the steps of:

retrieving a plurality of plans generated from a workflow; and

displaying each of the plans in a sequential manner to simulate the generation of the plans from the workflow; wherein said displaying is visually distinctive as a function of frequency of change in the plans.

29. The computer-readable medium of claim 28, farther comprising the steps of: receiving an indication of a rate of display; setting a time period equal to a reciprocal of the rate; and pausing for the time period before displaying each of the plans.

30. The computer-readable medium of claim 28, wherein the plans are displayed in reverse order.

31. The computer-readable medium of claim 28 wherein the display comprises a Gantt chart.

38. A data processing system comprising:

a secondary storage device further comprising a workflow;

a memory device further comprising a program that retrieves the workflow from the secondary storage device, that creates versions of the workflow, wherein each version reflects an instance in an edit history, that stores the versions of the workflow on the secondary storage device, and that displays the versions of the workflow in a sequential manner to simulate animation of the edit history; wherein said displaying is visually distinctive as a function of frequency of change in the edit history; and

a processor for running the program.

39. The data processing system of claim 38, wherein the program displays the versions of the workflow in reverse order.

40. The data processing system of claim 38, wherein the program displays the versions of the workflow as flow diagrams.

41. A data processing system comprising:

a secondary storage device further comprising a plan;

a memory device further comprising a program that retrieves the plan from the secondary storage device, that creates versions of the plan,

wherein each version reflects an instance in an edit history, that stores the versions of the plan on the secondary storage device, and that displays the versions of the plan in a sequential manner to simulate animation of the edit history; wherein said displaying is visually distinctive as a function of frequency of change in the edit history; and

a processor for running the program.

42. The data processing system of claim 41, wherein the program displays the versions of the plan in reverse order.

43. The data processing system of claim 41, wherein the program displays the versions of the plan as Gantt charts.

44. A system having versions of a plan, each reflecting an instance in an edit history, the system comprising:

means for storing indications of the versions of the plan; and

means for displaying the versions of the plan in a sequential manner to simulate animation of the edit history; wherein said displaying is visually distinctive as a function of frequency of change in the edit history.

9. Evidence Appendix

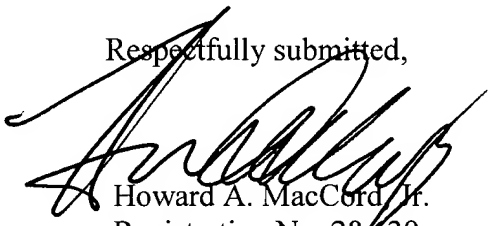
These references were cited by the Examiner in his rejections, and applicant relies on portion of them to show the errors of the rejections. Copies are attached.

Patent Number or Document Number	1st Named Inventor	Examiner Cited in Office Action Dated
U.S. 5,930,512	Boden et al.	June 1, 2005
U.S. 5,893,128	Nauckhoff	June 1, 2005
US 6,240,395	Kumashiro	June 1, 2005
US 5,845,279	Garofalakis et al.	June 1, 2005
Microsoft® Word 2000, "About adding comments and keeping track of changes," 1999, pp. 1-2 and Figure 1.		June 1, 2005
ls (2000, http://polyglotman.sourceforge.net/sgi-ls.l.html, page 7, line 2).		June 1, 2005

10. Related Proceedings Appendix

None.

Respectfully submitted,



Howard A. MacCord, Jr.
Registration No. 28,639
MacCord Mason PLLC
P. O. Box 2974
Greensboro, North Carolina 27402
(336) 273-4422.

Date: November 30, 2005
File No.: 7399-026

CERTIFICATE OF MAILING

I HEREBY CERTIFY THAT THIS DOCUMENT IS BEING DEPOSITED WITH THE UNITED STATES POSTAL SERVICE AS FIRST-CLASS MAIL, IN AN ENVELOPE ADDRESSED TO: **COMMISSIONER FOR PATENTS, P.O. BOX 1450, ALEXANDRIA, VA 22313-1450**, ON November 30, 2005

(Date of Deposit)

Christian E. Carter-Seyboth
Name of Depositor

Christian E. Carter-Seyboth
Signature

November 30, 2005
Date of Signature

About adding comments and keeping track of changes

This topic provides reference information about:

Using the Word change-tracking feature to track changes to a document

Inserting comments

Emphasizing text by highlighting it

Comparing documents

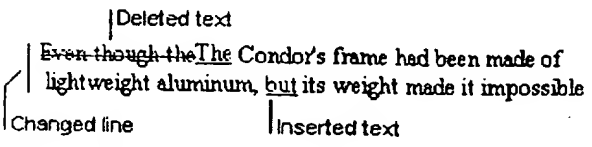
Merging tracked changes

Saving multiple versions of a document in one file

Using discussions to collaborate on a document

Using the Word change-tracking feature to track changes to a document

If you have a document that you want reviewed by your workgroup and you want final control over which changes to accept or reject, you can prepare copies of a document to distribute to the workgroup for electronic review and markup. When change tracking is turned on, Microsoft Word uses revision marks, the equivalent of "redlining" or "blacklining" in the legal profession, to indicate tracked changes. After the document is reviewed, you can see the changes made by the different reviewers — each reviewer's changes are marked with a different color.

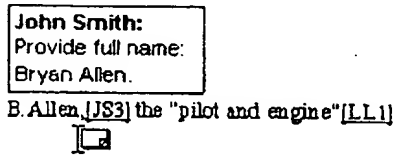


After viewing tracked changes, you can accept or reject each change. You can also choose to show or hide tracked changes on the screen or in the printed document by using the **Highlight Changes** dialog box (**Tools** menu, **Track Changes** submenu, **Highlight Changes** command). Track changes while you edit.

Return to top

Inserting comments

You and your reviewers can make annotations on the screen without changing the document text by typing comments by using **Insert Comment** on the **Reviewing** toolbar. When a comment is added, Word numbers it and records it in a separate comment pane. Word then inserts a comment reference mark in the document and shades the text that's commented on with light yellow. As with revision marks, Word tracks each reviewer's comment reference marks in a distinct color.



You can view comments in ScreenTips or in the comment pane. Use the comment pane to edit and review all the comments in the document. Insert a comment.

Return to top

Emphasizing text by highlighting it

When important text is highlighted, it's easier for you and your reviewers to see the text when you scan the document. You can change the color used for highlighting text and remove the highlighting. Highlight important text.

Return to top

Comparing documents

Comparing documents is useful for viewing changes to a document that was not revised by using change tracking. Open the newer version of the document, and then use the **Compare Documents** command (**Tools** menu, **Track Changes** submenu) to open the document you want to compare it with. When you compare documents, Word marks text that was added to and deleted from the old document, just as it would if change tracking had been turned on. Compare two copies of a document.

Return to top

Merging tracked changes

If reviewers have returned changes and comments in separate copies of the original document, you can merge all their changes into the original document and then review each change from there. Merge comments and changes from several reviewers into one document.

Return to top

Saving multiple versions of a document in one file

If you have a document that might be revised in the future, you might want to keep a record of each different draft. You can set up the document so that Word saves a "snapshot" of the current version automatically each time you or another reviewer closes the document after making changes. All versions are stored within one document, although Word displays only the current version by default. You save disk space because Word saves only the differences between versions, not an entire copy of each version.

For each version in the document, Word records the date and time the version was saved and the name of the person making the changes. You can view any one of the previous versions in a separate window by opening that version from the **Versions** dialog box (**File** menu). You can also open, print, and delete earlier versions. Automatically save a version when closing a document.

Compare saving multiple versions with saving a backup copy of a document, which is designed to ensure against data loss or against unintended changes. When backup saving is turned on, a new backup replaces the existing backup each time you save the document.

Return to top

Using discussions to collaborate on a document

If you're part of a corporate workgroup, and your server administrator has set up a discussion server for your workgroup, you and your co-workers can use the Discussions feature to share and collaborate on a Word document. Learn about using discussions in Word.

Return to top

Figure 1

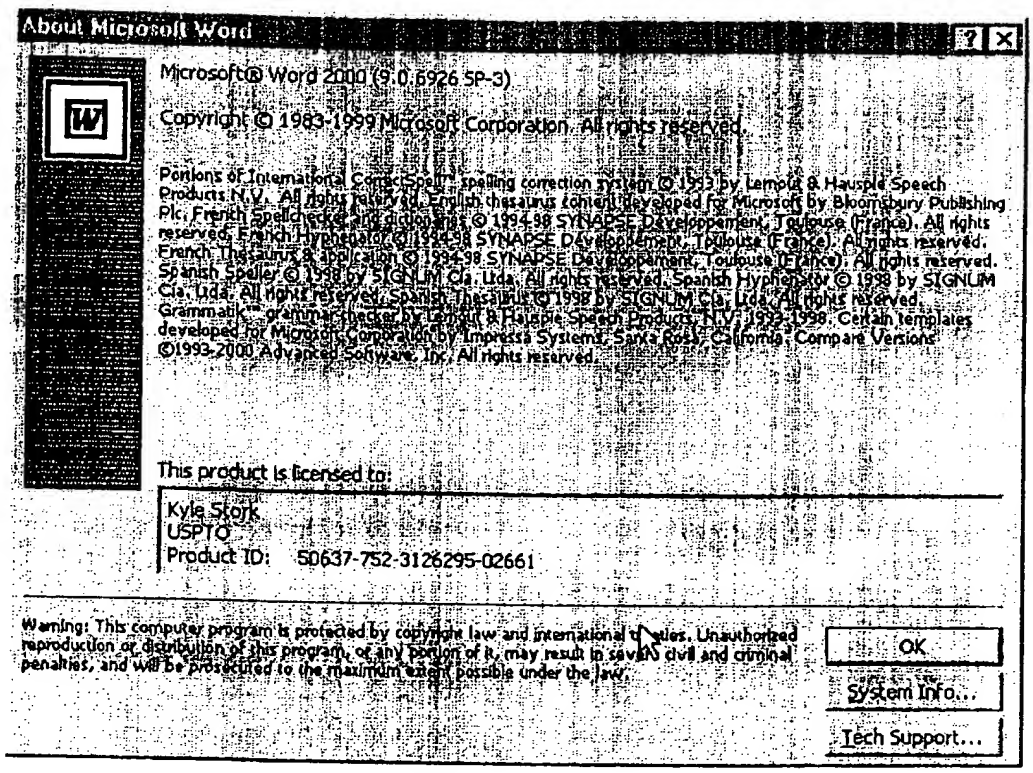

About Microsoft Word
Microsoft® Word 2000 (9.0.6926 SP-3)
Copyright © 1983-1999 Microsoft Corporation. All rights reserved.
Portions of International CorrectSpell™ spelling correction system © 1993 by Lernout & Hauspie Speech Products N.V. All rights reserved. English thesaurus content developed for Microsoft by Bloomsbury Publishing Plc. French spell checker and dictionary © 1994-98 SYNAPSE Developpement, Toulouse (France). All rights reserved. French Hyphenation © 1994-98 SYNAPSE Developpement, Toulouse (France). All rights reserved. French Thesaurus & Application © 1994-98 SYNAPSE Developpement, Toulouse (France). All rights reserved. Spanish Speller © 1998 by SIGNUM Cía. Ltda. All rights reserved. Spanish Hyphenator © 1998 by SIGNUM Cía. Ltda. All rights reserved. Spanish Thesaurus © 1998 by SIGNUM Cía. Ltda. All rights reserved. Grammatik™ grammar checker by Lernout & Hauspie Speech Products N.V. 1993-1998. Certain templates developed for Microsoft Corporation by Impressa Systems, Santa Rosa, California. Compare Versions ©1993-2000 Advanced Software, Inc. All rights reserved.
This product is licensed to:
Kyle Stork
USPTO
Product ID: 50637-752-3126295-02661
Warning: This computer program is protected by copyright law and international treaties. Unauthorized reproduction or distribution of this program, or any portion of it, may result in severe civil and criminal penalties, and will be prosecuted to the maximum extent possible under the law.
OK
System Info...
Tech Support...

Table of Contents

NAME

ls - list contents of directory

SYNOPSIS

ls [-RaAdCLHxmlnogrtucpFbqisf] [names]

DESCRIPTION

For each directory argument, *ls* lists the contents of the directory; for each file argument, *ls* repeats its name and any other information requested. The output is sorted alphabetically by default. When no argument is given, the current directory is listed. When several arguments are given, the arguments are first sorted appropriately, but file arguments appear before directories and their contents.

There are three major listing formats. The default format is to list one entry per line, the **-C** and **-x** options enable multi-column formats, and the **-m** option enables stream output format. In order to determine output formats for the **-C**, **-x**, and **-m** options, *ls* uses an environment variable, **COLUMNS**, to determine the number of character positions available on one output line. If this variable is not set, the terminfo(4) database is used to determine the number of columns, based on the environment variable **TERM**. If this information cannot be obtained, 80 columns are assumed.

The *ls* command has the following options:

-R

Recursively list subdirectories encountered.

-a

List all entries, including those that begin with a dot (.), which are normally not listed.

-A

Like **-a** except it does not list the . and .. directories.

-d

If an argument is a directory, list only its name (not its contents); often used with **-l** to get the status of a directory.

-C

Multi-column output with entries sorted down the columns.

-L

If the file is a symbolic link, list the file that the link references.

-H

If the file is a symbolic link, list the file itself. This is the default behavior, except for a symbolic link to a directory in the absence of the **-l** option, in which case the -L behavior is used by default. **-H** and **-L** cancel one another in command line order.

-x

Multi-column output with entries sorted across rather than down the page.

-m

Stream output format; files are listed across the page, separated by commas.

-l

List in long format, giving mode, number of links, owner, group, size in bytes, and time of last modification for each file (see below). If the file is a special file, the size field will instead contain the major and minor device numbers rather than a size.

-n

The same as **-l**, except that the owner's **UID** and group's **GID** numbers are printed, rather than the associated character strings.

-o

The same as **-l**, except that the group is not printed.

-g

The same as **-l**, except that the owner is not printed.

-r

Reverse the order of sort to get reverse alphabetic or oldest first as appropriate.

-t

Sort by time stamp (latest first) instead of by name. The default is the last modification time. (See **-u** and **-c**.)

-u

Use time of last access instead of last modification for sorting (with the **-t** option) or printing (with the **-l** option).

-c

Use time of last modification of the i-node (file created, mode changed, etc.) for sorting (**-t**) or printing (**-l**).

-p

Put a slash (/) after the name of each file that is a directory.

-F

Put a slash (/) after the name of each file that is a directory, put an asterisk (*) after the name of each file that is executable, put a commercial at sign (@) after the name of each file that is a symbolic link and put an equals sign (=) after the name of each file that is a AF_UNIX address family socket.

-b

Force printing of non-printable characters to be in the octal **\\ddd** notation.

-q

Force printing of non-printable characters in file names as the character question mark (**?**).

-i

For each file, print the i-number in the first column of the report.

-s

Give size in blocks, including indirect blocks, for each entry.

-f

Force each argument to be interpreted as a directory and list the name found in each slot. This option turns off **-l, -t, -s,** and **-r,** and turns on **-a;** the order is the order in which entries appear in the directory.

The mode printed under the **-l** option consists of ten characters. The first character may be one of the following:

d

the entry is a directory;

b

the entry is a block special file;

c

the entry is a character special file;

l

the entry is a symbolic link;

p

the entry is a fifo (a.k.a. "named pipe") special file;

s

the entry is a AF_UNIX address family socket;

-

the entry is an ordinary file.

The next 9 characters are interpreted as three sets of three bits each. The first set refers to the owner's permissions; the next to permissions of others in the user-group of the file; and the last to all others. Within each set, the three characters indicate permission to read, to write, and to execute the file as a program, respectively. For a directory, "execute" permission is interpreted to mean permission to search the directory for a specified file.

ls -l (the long list) prints its output as follows:

-rwxrwxrwx

1 smith dev 10876 May 16 9:42 part2

This horizontal configuration provides a good deal of information. Reading from right to left, you see that the current directory holds one file, named "part2." Next, the last time that file's contents were modified was 9:42 A.M. on May 16. The file is moderately sized, containing 10,876 characters, or bytes. The owner of the file, or the user, belongs to the group "dev" (perhaps indicating "development"), and his or her login name is "smith." The number, in this case "1," indicates the number of links to file "part2." Finally, the row of dash and letters tell you that user, group, and others have permissions to read, write, execute "part2."

The execute (**x**) symbol here occupies the third position of the threecharacter sequence. A - in the third position would have indicated a denial of execution permissions.

The permissions are indicated as follows:

r
: the file is readable

w
: the file is writable

x
: the file is executable

-
: the indicated permission is *not* granted

l
: mandatory locking will occur during access (the set-group-ID bit is on and the group execution bit is off)

s
: the set-user-ID or set-group-ID bit is on, and the corresponding user or group execution bit is also on

S
: undefined bit-state (the set-user-ID bit is on and the user execution bit is off)

t

the 1000 (octal) bit, or sticky bit, is on (see chmod(1)), and execution is on

T

the 1000 bit is turned on, and execution is off (undefined bitstate)

For user and group permissions, the third position is sometimes occupied by a character other than **x** or -. **s** also may occupy this position, referring to the state of the set-ID bit, whether it be the user's or the group's. The ability to assume the same ID as the user during execution is, for example, used during login when you begin as root but need to assume the identity of the user stated at "login."

In the case of the sequence of group permissions, **l** may occupy the third position. **l** refers to mandatory file and record locking. This permission describes a file's ability to allow other files to lock its reading or writing permissions during access.

For others permissions, the third position may be occupied by **t** or **T**. These refer to the state of the sticky bit and execution permissions.

EXAMPLES

An example of a file's permissions is:

-rwxr--r-

This describes a file that is readable, writable, and executable by the user and readable by the group and others.

Another example of a file's permissions is:

-rwsr-xr-x

This describes a file that is readable, writable, and executable by the user, readable and executable by the group and others, and allows its user-ID to be assumed, during execution, by the user presently executing it.

Another example of a file's permissions is:

-rw-rwl--

This describes a file that is readable and writable only by the user and the group and can be locked during access.

An example of a command line:

ls -a

This command will print the names of all files in the current directory, including those that begin with a dot (.), which normally do not print.

Another example of a command line:

ls -aisn

This command will provide you with quite a bit of information including **a**ll files, including non-printing ones (**a**), the **i**-number-the memory address of the i-node associated with the file-printed in the left-hand column (**i**); the **s**ize (in blocks) of the files, printed in the column to the right of the i-numbers (**s**); finally, the report is displayed in the **n**umeric version of the long list, printing the **UID** (instead of user name) and **GID** (instead of group name) numbers associated with the files.

When the sizes of the files in a directory are listed, a total count of blocks, including indirect blocks, is printed.

FILES

/etc/passwd
    user IDs for **ls -l** and **ls -o**

/etc/group
    group IDs for **ls -l** and **ls -g**

/usr/lib/terminfo/?/*
    terminal information database

SEE ALSO

chmod(1), find(1).

BUGS

Unprintable characters in file names may confuse the columnar output options.

Table of Contents